DETAILED ACTION

1. This action is responsive to the amendment filed March 24, 2008.

2. Per applicants' request, claims 1-6, 9-16, 19,21, 24-28, 30,32, and 36 have been amended and claim 35 has been cancelled. Claim 37 has been newly introduced.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 1-4 and 37 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Regarding claims 1 and 37, where Applicant discloses "identifying the first hot translated block as associated with a hot loop when the hot execution trip count exceeds a second threshold" is not fully supported inside the Specification. On page 17, paragraph 0057, Applicant states that if a *prefetch candidate's trip_count* exceeds the second predetermined threshold then the loop will be identifies as a hot loop not if the hot

execution trip count exceeds a second threshold. Thus, the claim is rejected under 35 U.S.C. 112 1st Paragraph.

The rejection of the independent claims are incorporated into their dependent claims

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-10, 21-34, 36 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Roediger (U.S. Pat 6,938,249) and further in view of Santhanam. (U.S. Pat 5,704,053)

With respect to claim 1 and 37, Roediger discloses a machine readable medium(Col 3:44-55, "…compiling system…") storing instructions structured to cause a machine to: cold translating a plurality of blocks of the program(Col 5:35-45, '…The blocks of code are basic blocks…") from a first language to a second language to generate a cold translated program; (Col 3:44-60, "…Front-end compilers are used to convert …")determine a cold execution trip count associated with a first one of the blocks in the

cold translated program;(Col 5:45-60, "…once the average number of loop executions per loop entry has been computed…to peel or unroll a loop…")

Identifying the first block for hot translation when the cold execution trip count exceeds a first threshold;(Col 5:50-67, "…uses an arbitrary peeling threshold of two executions per loop entry to decide whether to peel a loop…."; col. 7, line 40 – col. 8, line 53)

When the first block is identified for hot translation, hot translating the first block into a first hot translated block by inserting a hot execution trip counting instruction(Col 5:45-55, "…for determining the average number of executions per loop entry…"; col. 7, line 40 – col. 8, line 53) into a first hot translated block to calculate a hot execution trip count for the first hot translated block; (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…"; col. 7, line 40 – col. 8, line 53)

Linking the first hot translated block into the cold translated program; (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…")

Executing the cold translated program with the first hot translated block; (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…"; col. 7, line 40 – col. 8, line 53)

Identifying the first hot translated block as associated with a hot loop when the hot execution trip count exceeds a second threshold;(Col 6:40-55, "…If the leftover value is zero…If the leftover value is two…"; col. 7, line 40 - col. 8, line 53)

inserting instrumentation into the hot loop to develop profile data for a load instruction within the instrumented hot loop; (Col 7:24-55, “…inserts additional instrumentation code into an instruction stream to collect the more detailed profile data…”)
linking the instrumented hot translated block into the cold translated program;
executing the cold translated program with the instrumented hot loop;(Col 8:10-30, "...to optimize one or more loops according to the profile data...") but does not disclose and inserting a prefetching instruction into the hot loop if the profile data indicates the load instruction in the instrumented hot loop meets a predefined criteria, the prefetching instruction to prefetch data for the load instruction.

Santhanam discloses inserting a prefetching instruction into the hot loop if the profile data indicates the load instruction in the instrumented hot loop meets a predefined criteria,(Col 15:30-43, “…if M’ is <= cache line size, then a prefetch strategy suited to small strides is employed…”; abstract) the prefetching instruction to prefetch data for the load instruction. (Col 21:10-40, “…and target register of the load, then one or more prefetch instructions are inserted into the code…"; abstract) in an analogous system for the purpose of eliminating cache miss overhead during program execution at runtime.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to insert prefetch instructions that pre-load data for load instructions inside program loops during compilation.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to eliminate cache miss overhead during program execution at runtime.

With respect to claim 5 and 25, Roediger discloses an apparatus having a logic circuit,(Col 13:15-25, "…one or more microprocessors and/or integrated circuits…") the apparatus to optimize a program comprising: cold translating a plurality of blocks of the program(Col 5:35-45, "…The blocks of code are basic blocks…") from a first language to a second language to generate a cold translated program; (Col 3:44-60, "…Front-end compilers are used to convert …")determine a cold execution trip count associated with a first one of the blocks in the cold translated program;(Col 5:45-60, "…once the average number of loop executions per loop entry has been computed…to peel or unroll a loop…" col. 7, line 40 – col. 8, line 53)
Identifying the first block for hot translation when the cold execution trip count exceeds a first threshold;(Col 5:50-67, "…uses an arbitrary peeling threshold of two executions per loop entry to decide whether to peel a loop…." col. 7, line 40 – col. 8, line 53)
When the first block is identified for hot translation, hot translating the first block into a first hot translated block by inserting a hot execution trip counting instruction(Col 5:45-55, "…for determining the average number of executions per loop entry…") into a first hot translated block to calculate a hot execution trip count for the first hot translated block; (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…" col. 7, line 40 – col. 8, line 53)

A code linker to link the at least one hot translated block into the cold translated program to generate a hot translated program; (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…")

Executing the cold translated program with the first hot translated block; (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…") Identifying the first hot translated block as associated with a hot loop when the hot execution trip count exceeds a second threshold;(Col 6:40-55, "…If the leftover value is zero…If the leftover value is two…" col. 7, line 40 – col. 8, line 53)

A gen-translator to instrument the hot loop with instructions to collect profile information; (Col 7:24-55, "…inserts additional instrumentation code into an instruction stream to collect the more detailed profile data… col. 7, line 40 – col. 8, line 53")but does not disclose

and a use-translator to insert a prefetch instruction associated with the hot loop based on the profile information.

Santhanam discloses a use-translator to insert a prefetch instruction associated with the hot loop based on the profile information(Col 15:30-43, "…if M' is <= cache line size, then a prefetch strategy suited to small strides is employed…"; abstract) in an analogous system for the purpose of eliminating cache miss overhead during program execution at runtime.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to insert prefetch instructions that pre-load data for load instructions inside program loops during compilation.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to eliminate cache miss overhead during program execution at runtime.

Dependent claims

With respect to claim 2, the rejection of claim 1 is incorporated and further, Santhanam discloses that inserting instrumentation into the loop comprises: finding the load instruction in the hot loop; (Col 21:10-40, "…and target register of the load, then one or more prefetch instructions are inserted into the code…")

and inserting a first instruction sequence to record addresses associated with the load instruction. (Col 21:10-40, "…and target register of the load, then one or more prefetch instructions are inserted into the code…")

With respect to claim 3, the rejection of claim 2 is incorporated and further, Santhanam discloses that the first instruction sequence causes the addresses to be recorded in a buffer associated with the loop, and inserting instrumentation into the loop further comprises: inserting a second instruction sequence into the loop to trigger processing of

the addresses in the buffer to determine if the profile data indicates a load instruction in the loop meets a predefined criteria. (Col 15:30-43, "…if M' is <= cache line size, then a prefetch strategy suited to small strides is employed…")

With respect to claim 4, the rejection of claim 1 is incorporated and further, Santhanam discloses that profile data identifies the load instruction as at least one of a single stride load, a multiple stride load, a cross stride load, and a base load of a cross stride load. (Col 15:30-43, "…if M' is <= cache line size, then a prefetch strategy suited to small strides is employed…")

With respect to claim 6, the rejection of claim 5 is incorporated and further, Roediger discloses that cold translating the program comprises: identifying a block in a foreign program; inserting instructions to update a first counter into an instruction block to determine the number of times the instruction block is executed;(Col 5:30-45, "…First the total number of loop executions are determined…" col. 7, line 40 – col. 8, line 53) and analyzing the first counter to determine if the block is a candidate for optimization. (Col 5:55-67, "…peeling threshold of two executions per loop entry to decide whether to peel a loop…" col. 7, line 40 – col. 8, line 53)

With respect to claim 7, the rejection of claim 5 is incorporated and further, Roediger discloses that gen-translating and use-translating the program each comprises translating the first instruction set to an intermediate instruction set (Col 3:44-60,

"…Front-end compilers are used to convert …") and translating the intermediate instruction set to the second instruction set. (Col 3:44-60, "…Back-end compilers are used to convert the intermediate representation to machine code…")

With respect to claim 8, the rejection of claim 7 is incorporated and further, Roediger discloses that the intermediate instruction set comprises an instruction set different than the first instruction set and different than the second instruction set. (Col 3:44-60, "…Front-end compilers are used to convert …")

With respect to claim 9, the rejection of claim 5 is incorporated and further, Roediger discloses that identifying the hot loop in the translated program comprises conditioning a loop by a least common specialization operation. (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…" col. 7, line 40 – col. 8, line 53)

With respect to claim 10, the rejection of claim 9 is incorporated and further, Roediger discloses that the least common specialization operation comprises: identifying a block of instructions associated with the hot loop that is a least common denominator block with other loops; (Col 5:33-45, "…the first basic block is entered 100 times…")
rotating the loop such that the least common denominator block is a head of the loop. (Col 5:33-45, "…the first basic block is entered 100 times…")

With respect to claim 21, the rejection of claim 5 is incorporated and further, Santhanam discloses that use-translating comprises: analyzing the profile information; (Col 15:30-45, '…determining the fewest number of prefetch instructions needed to ensure full cache miss coverage…")
and inserting a prefetching instruction for the load instruction. (Col 21:10-35, '…insert as many prefetches as needed to cover the cluster's entire span…")

With respect to claim 22, the rejection of claim 21 is incorporated and further, Santhanam discloses further comprising eliminating redundant prefetched loads. (Col 21:10-40, "…insert as many prefetches as needed to cover the cluster's entire span…")

With respect to claim 23, the rejection of claim 21 is incorporated and further, Santhanam discloses that analyzing the profile information comprises determining if the load instruction is at least one of: a single stride load, a multiple stride load, a cross

stride load; and a base load. (Col 15:30-43, "…if M' is <= cache line size, then a prefetch strategy suited to small strides is employed…")

With respect to claim 24, the rejection of claim 5 is incorporated and further, Santhanam discloses further comprising linking the use-translated hot loop into the native program. Col 4:55-67, "…and linker so that when it is run on the computer…")

With respect to claim 26, the rejection of claim 25 is incorporated and further, Roediger discloses that the hot loop identifier identifies a loop as a hot loop by: counting a number of times an instruction block associated with the loop is executed; (Col 5:45-60, basic block is entered 100 times…") determining an average number of times the loop is executed; (Col 5:45-60, "…determining the average number of executions per loop entry…")
and comparing the average number of times the loop is executed to a predetermined threshold. (Col 5:55-67, "...uses an arbitrary peeling threshold...")

With respect to claim 27, the rejection of claim 25 is incorporated and further, Roediger discloses that the hot loop identifier identifies a hot loop in the translated program by conditioning a loop by a least common specialization operation. (Col 6:23-40, "…the average executions per entry for this loop is ten…the loop should be unrolled…")

With respect to claim 28, the rejection of claim 27 is incorporated and further, Roediger

discloses that the least common specialization operation comprises: identifying a block of instructions that is a least common denominator block with other loops; (Col 5:33-45, "…the first basic block is entered 100 times…")

rotating the loop such that the least common denominator block is a head of the loop. (Col 5:33-45, "…the first basic block is entered 100 times…")

With respect to claim 29, the rejection of claim 25 is incorporated and further, Roediger discloses that the gen-translator and the use-translator each translates the program from the first instruction set to an intermediate instruction set (Col 3:44-60, "…Front-end compilers are used to convert …") and from the intermediate instruction set to the second instruction set. (Col 3:44-60, "…Back-end compilers are used to convert the intermediate representation to machine code…")

With respect to claim 30, the rejection of claim 25 is incorporated and further, Santhanam discloses that the gen-translator comprises:

a load instruction identifier to identify a load instruction within the hot loop and having at least one predetermined characteristic; (Col 21:15-45, '…if the memory reference corresponding to a cluster leader…where "disp" is a displacement…") a profiler to insert profiling instructions into the hot loop if the load instruction identifier identifies a load instruction within the hot loop having the at least one predetermined characteristic. ((Col 21:10-40, "…and target register of the load, then one or more prefetch instructions are inserted into the code…")

With respect to claim 31, the rejection of claim 30 is incorporated and further, Santhanam discloses that the profiler collects stride information for the load instruction. (Col 20:30-45, "…the stride required to achieve an overlap of C(i) with C(j),….")

With respect to claim 32, the rejection of claim 25 is incorporated and further, Roediger discloses that the use-translator comprises: a profile analyzer to determine a load instruction type for the load instruction based on the profile data; (Col 12:1-35, "…Profile data is data collected…")Santhanam discloses
an optimizer to insert a prefetch instruction into the loop for the load instruction;(Col 21:10-40, "…insert as many prefetches as needed…") and a code linker to couple the hot loop to the program. (Col 4:55-67, "…and linker so that when it is run on the computer…")

With respect to claim 33, the rejection of claim 32 is incorporated and further, Santhanam discloses that the optimizer determines an address to be prefetched based on the load instruction type. (Col 22:20-40, "…if the prefetch instruction supports an addressing mode…")

With respect to claims 34 and 36, the rejection of claim 32 and 35 are incorporated respectively and further, Santhanam discloses that the load instruction type comprises at least one of: a single stride load, a multiple stride load, a cross stride load, and a

base load of a cross stride load. ((Col 15:30-43, "…if M' is <= cache line size, then a prefetch strategy suited to small strides is employed…")

8. Claims 11-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Roediger (U.S. Pat 6,938,249) and in view of Santhanam. (U.S. Pat 5,704,053) and further in view of Granston.(U.S. PGPUB 2003/0140334)

With respect to claim 11, the rejection of claims 5 is incorporated and further,

Neither Roediger nor Santhanam disclose, that identifying the hot loop in the translated program comprises: using at least one of a cold execution trip count to determine the average number of times the hot loop is executed during cold execution or a hot execution trip count to determine the number of times the hot loop is executed.

Granston discloses that identifying the hot loop in the translated program comprises: using at least one of a cold execution trip count to determine the average number of times the hot loop is executed during cold execution(Col 7:0100, "…to decide if a trip count is less than two…") or a hot execution trip count to determine the number of times the hot loop is executed in an analogous system for the purpose of providing additional optimizing information to the compiler.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include a cold execution trip count to determine the average number of hot loop execution using the teachings of Granston.

With respect to claim 12, Granston discloses that the cold trip count comprises instructions to determine the frequency a loop entry block is taken and the frequency the loop back edge is taken. (Col 5:0067-0068, "…one for the case where the trip count is above a certain value…")

With respect to claim 13, Granston discloses that the cold trip count comprises instructions that the hot loop is gen-translated if the hot loop contains a load instruction and a value of at least one of a hot trip count and a cold trip count is greater than a predetermined threshold. (Col 5:0067-0068, "…one for the case where the trip count is above a certain value…")

With respect to claim 14, Granston discloses that the hot loop is only gen-translated if the load instruction does not access data in a stack or have a loop invariant load address. (Col 7:0100-0101, "…to decide if a trip count…")

With respect to claim 15, Granston discloses that the hot loop is optimized by a normal

hot translation if the cold trip count is less than the predetermined threshold. (Col 7:0100-0101, "…to decide if a trip count…")

With respect to claim 16, the rejection of claim 5 is incorporated and further,

Roediger nor Santhanam does not disclose that gen-translating comprises: identifying a load instruction within the hot loop; inserting a profiling instruction in association with the load instruction; inserting a profiling control instruction in a loop entry block of the loop to control the number of times the load instruction is profiled; executing the profiling instruction to profile the load instruction; and executing the profiling control instruction to determining if the load has been profiled more than a predetermined number of times.

Granston discloses that gen-translating comprises: identifying a load instruction within the hot loop; (Col 3:0053-0055, "…user loads a saved profile…")
inserting a profiling instruction in association with the load instruction;(Col 4:0066, "…or load a profile dataset…")
inserting a profiling control instruction in a loop entry block of the loop to control the number of times the load instruction is profiled; (Col 5:0068-0069, "…The loop identifier…to collect run time loop profile information…")
executing the profiling instruction to profile the load instruction; (Col 5:0068-0069, "…to collect run time loop profile information…")

and executing the profiling control instruction to determining if the load has been profiled more than a predetermined number of times(Col 5:0067, "…This extra loop is referred …") in an analogous system for the purpose of providing additional information to the compiler for optimization of loops.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to insert a profiling instruction in association with a load instruction using the teachings of Granston.

With respect to claim 17, the rejection of claim 16 is incorporated and further, Granston discloses that the profiling instruction comprises an instruction to assign the load instruction a unique identification number and an instruction to collect profiling information. (Col 5:0068-0069, "…The unique identifiers provided by the profile server…")

With respect to claim 18, the rejection of claim 17 is incorporated and further, Granston discloses that the unique identification number is stored with a data address of the load instruction. (Col 5:0068-0069, "…The unique identifiers provided by the profile server…")

With respect to claim 19, the rejection of claim 16 is incorporated and further, Granston discloses that the profiling information comprises stride information. (Col 5:0069, "…The

dataset identifier...")

With respect to claim 20, the rejection of claim 16 is incorporated and further, Granston discloses that the profiling control instruction comprises a counter to determine how many times the load instruction has been profiled. (Col 5:0068, "...record the loop identifier...")

Response to Arguments

9. Applicant's arguments with respect to claim1-34 and 36-37 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MARK P. FRANCIS whose telephone number is (571)272-7956. The examiner can normally be reached on Mon-Fri 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Mark P. Francis

Patent Examiner

Art Unit 2193

/Lewis A. Bullock, Jr./

Supervisory Patent Examiner, Art Unit 2193